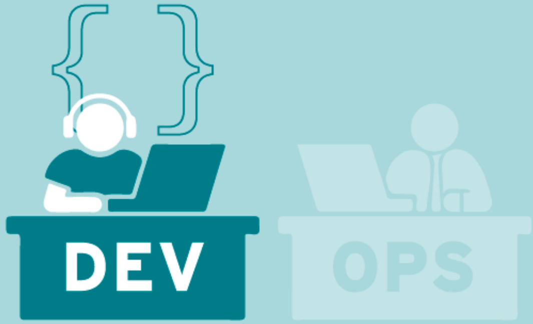
Red Hat



The Containers and Cloud-Native Roadshow Developer Track Lab Guide

A hands-on experience for Ops and Dev professionals

DEVELOPER TRACK MODULES

1 OPTIMIZING EXISTING APPLICATIONS

Migrate an existing monolithic Java application from a legacy platform to Red Hat.

Modernize by incrementally refactoring to microservices architecture and modern Java platform

2 DEBUGGING, MONITORING AND CONTINUOUS DELIVERY

Debug, instrument and monitor a modern microservice application.

Deploy continuously using Pipelines

3 CONTROL CLOUD NATIVE APPS WITH SERVICE MESH

Gain a deep understanding of app behavior through service mesh instrumentation and visualization

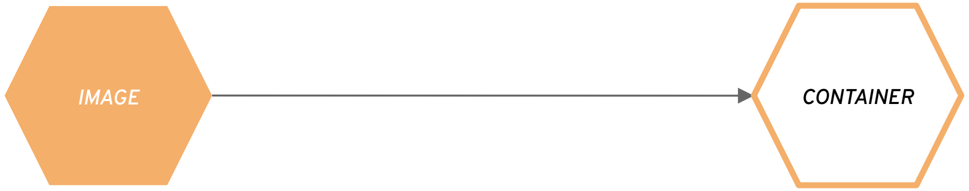
4 ADVANCED CLOUD NATIVE WITH EVENT-DRIVEN SERVERLESS

Dynamically respond to events and scale applications using powerful Kubernetes constructs

OpenShift Concepts

a container is the smallest compute unit

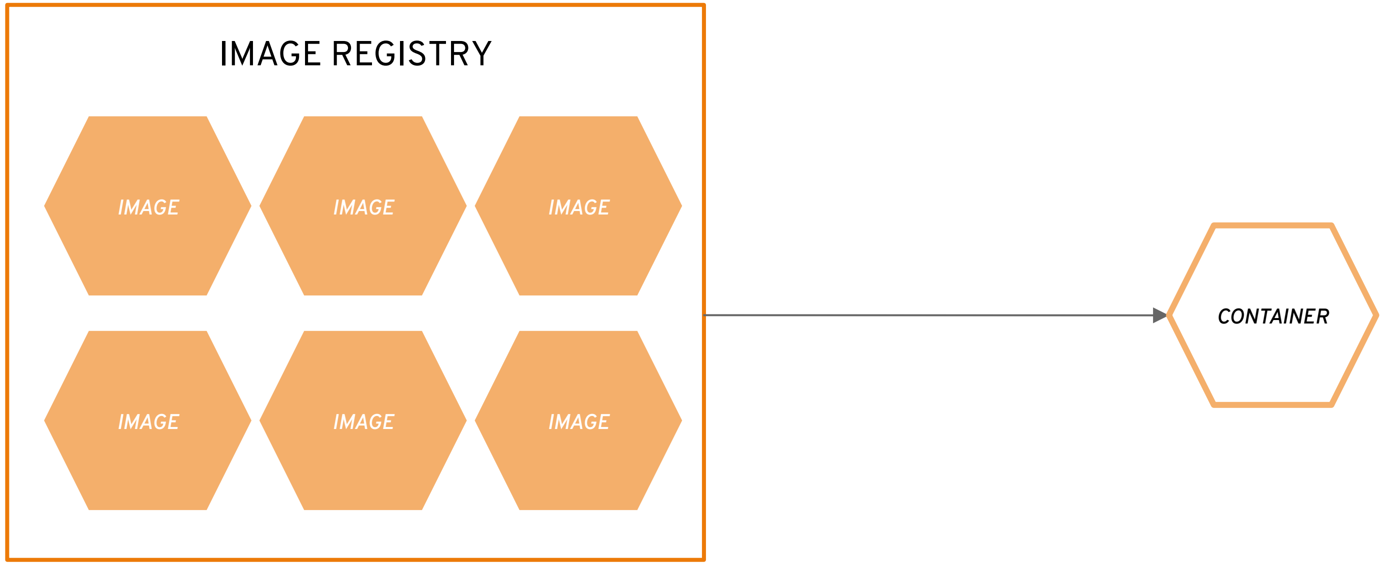
containers are created from container images



BINARY

RUNTIME

container images are stored in an image registry



an image repository contains all versions of an image in the image registry

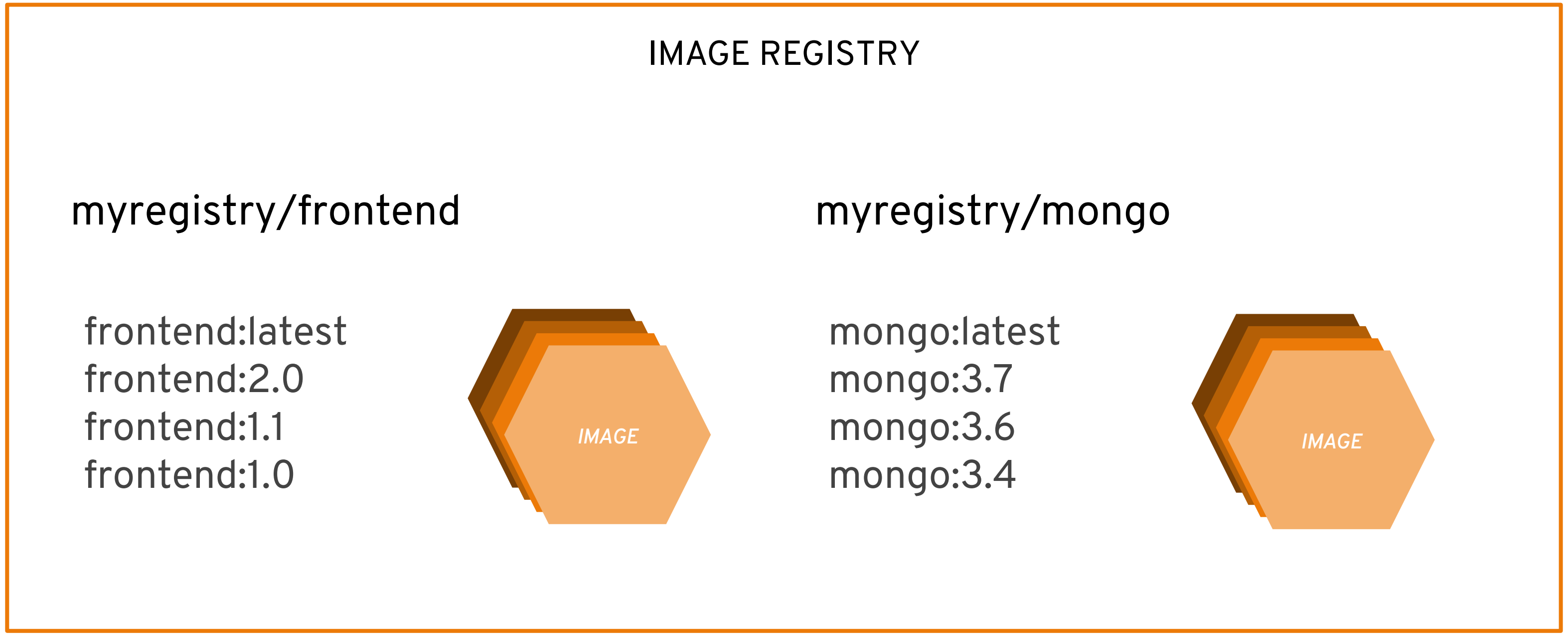


IMAGE REGISTRY

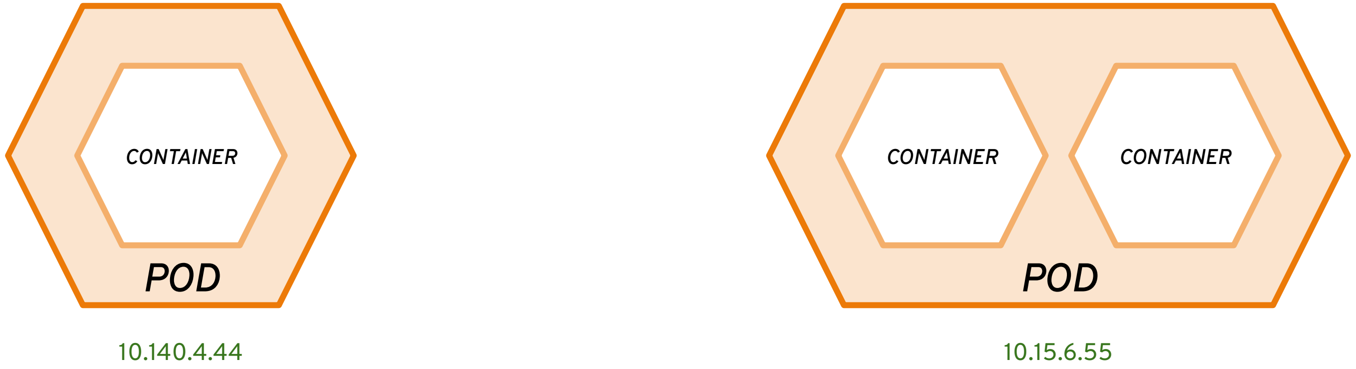
myregistry/frontend

frontend:latest
frontend:2.0
frontend:1.1
frontend:1.0

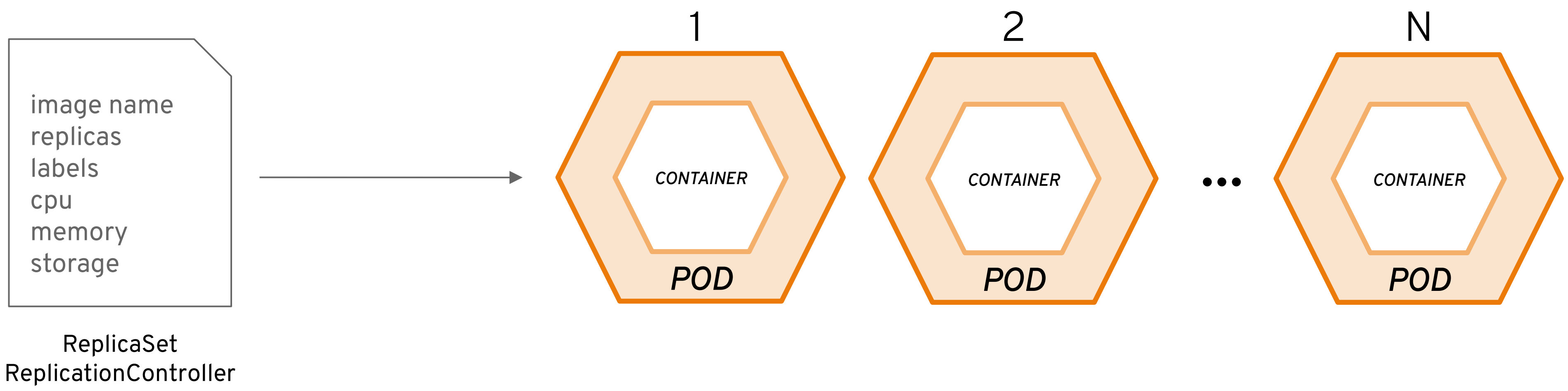
myregistry/mongo

mongo:latest
mongo:3.7
mongo:3.6
mongo:3.4

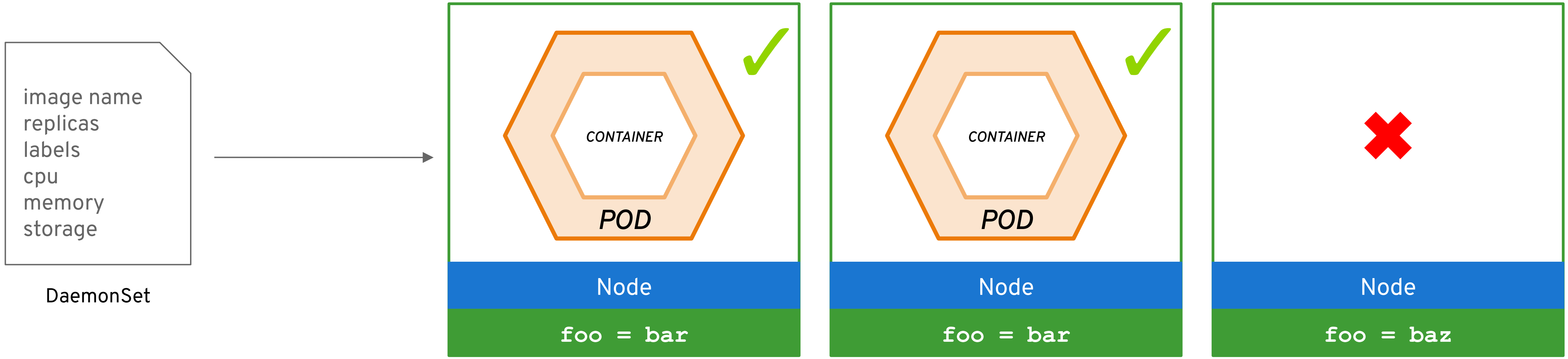
containers are wrapped in pods which are units of deployment and management



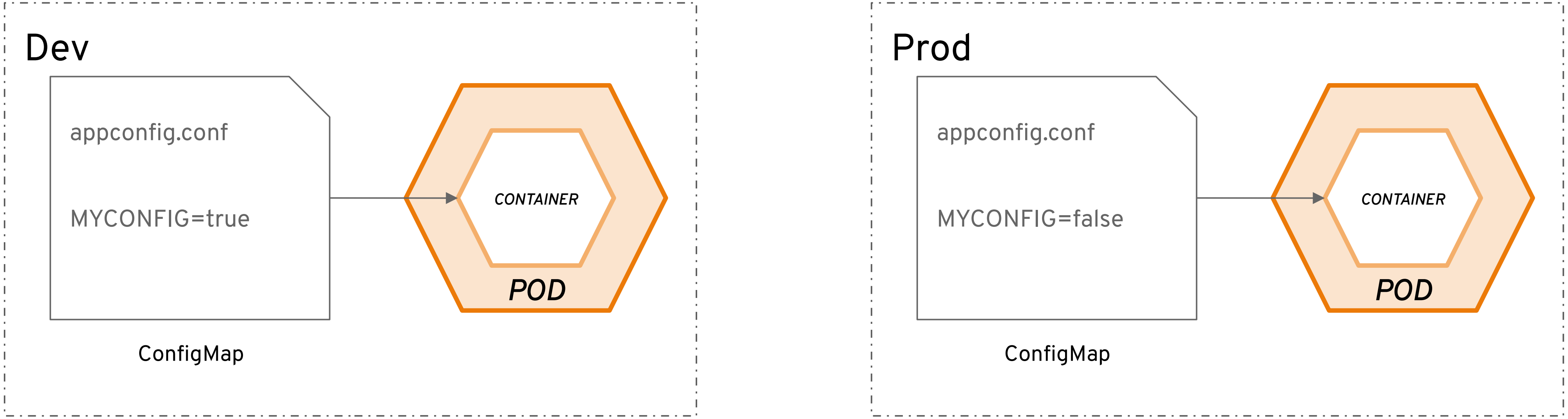
`ReplicationControllers` & `ReplicaSets` ensure a specified number of pods are running at any given time



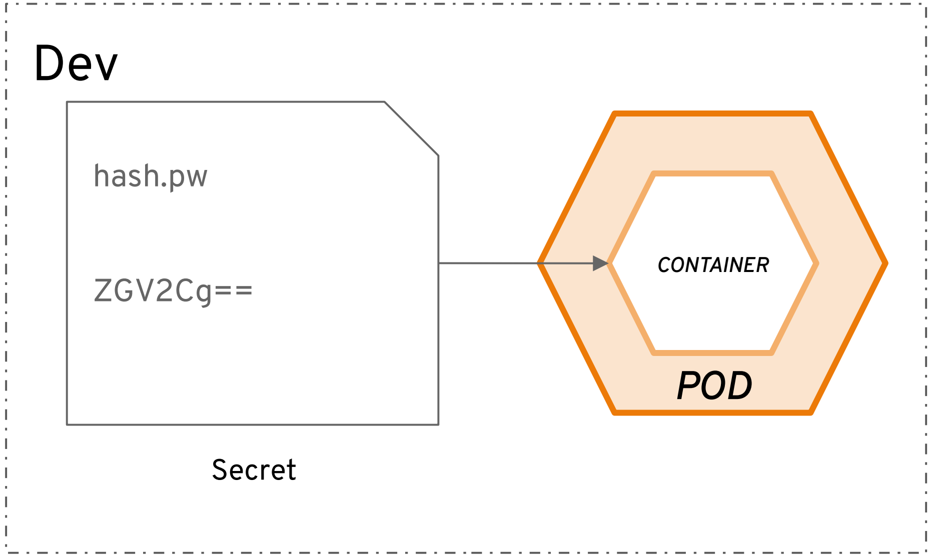
a `daemonset` ensures that all (or some) nodes run a copy of a pod

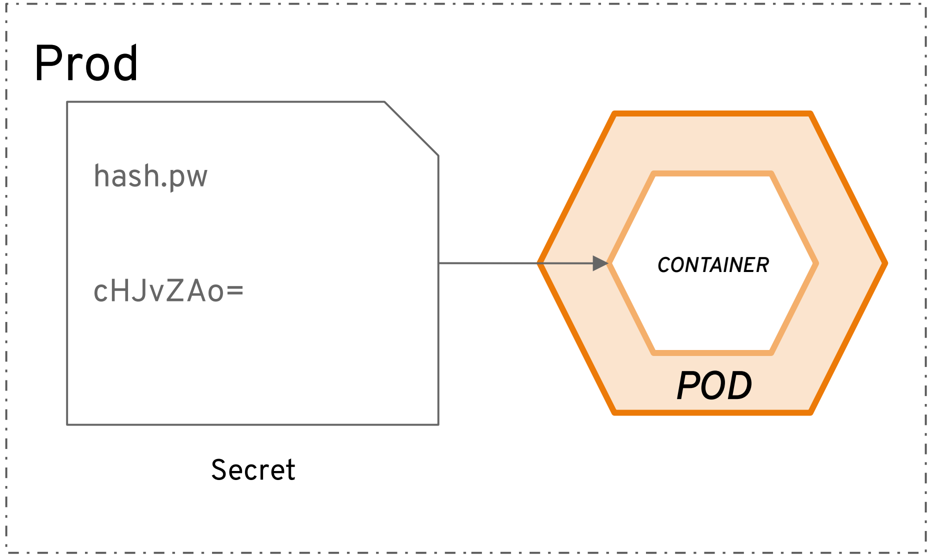


`configmaps` allow you to decouple configuration artifacts from image content

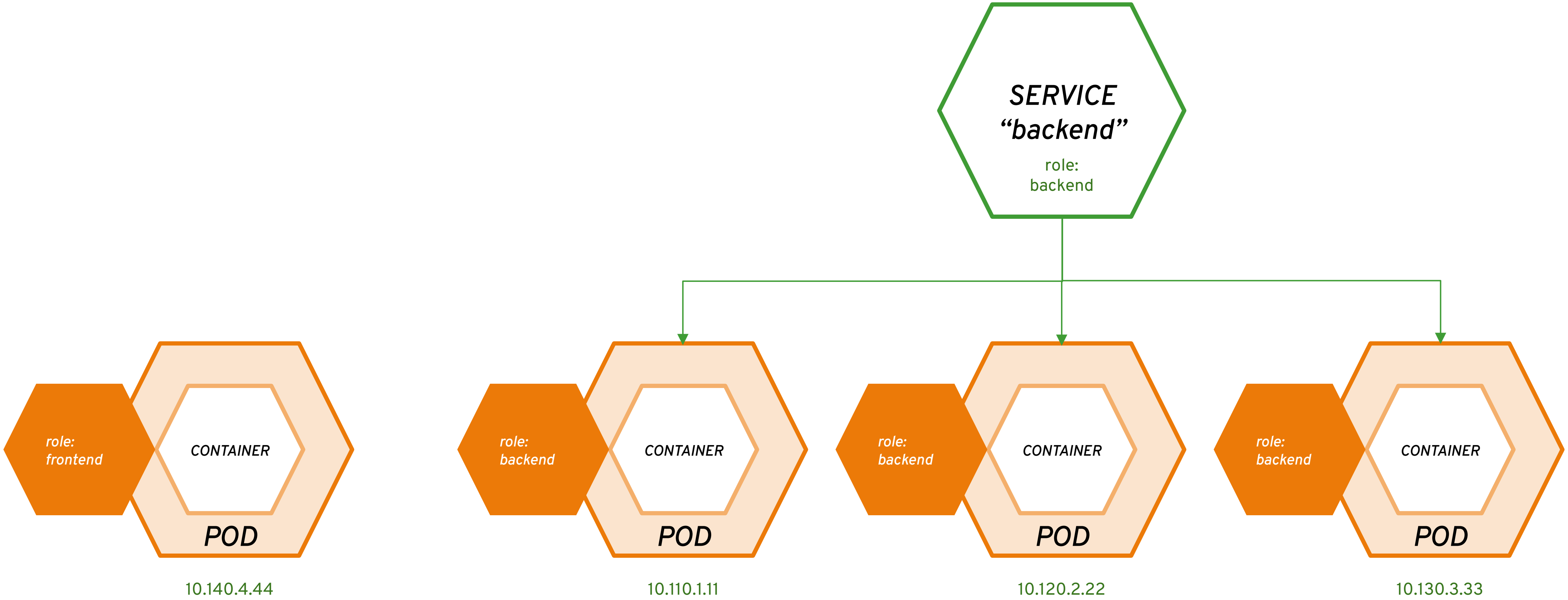


`secrets` provide a mechanism to hold sensitive information such as passwords

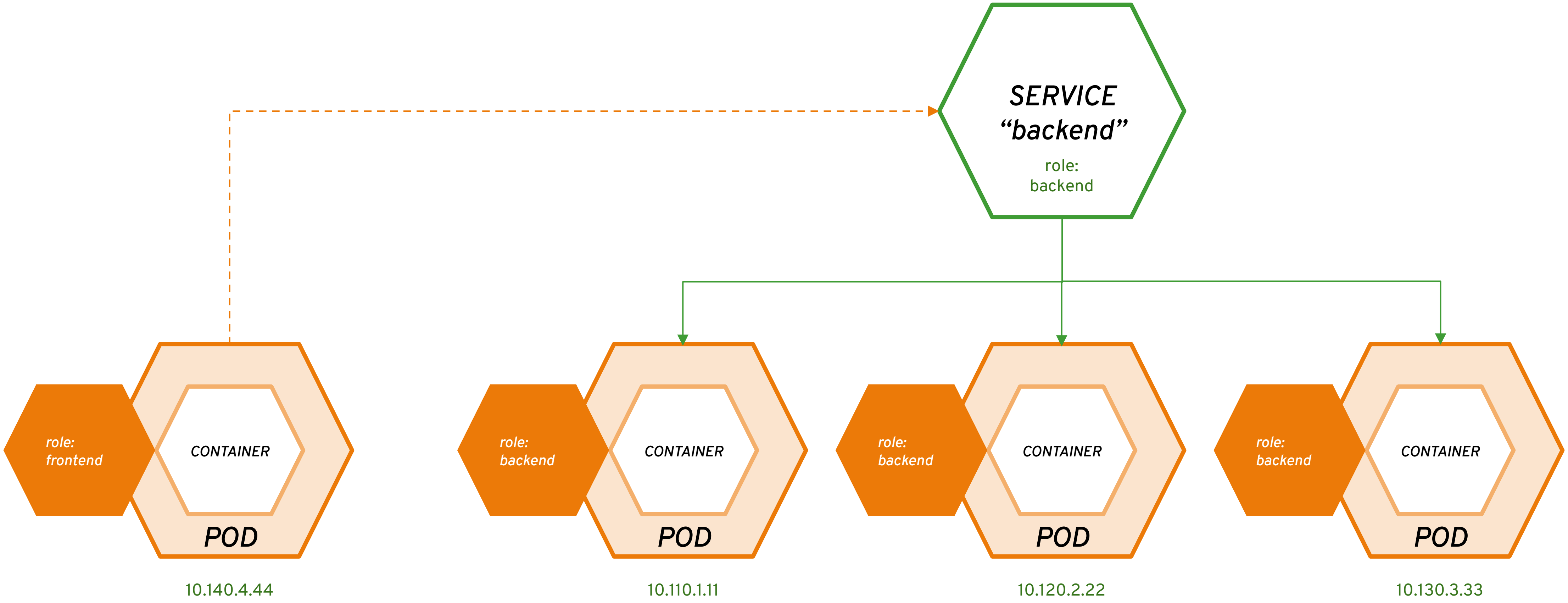




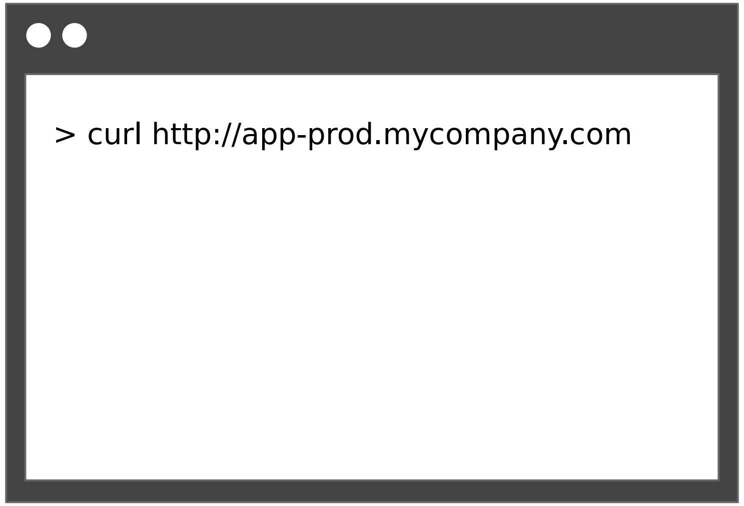
services provide internal load-balancing and service discovery across pods



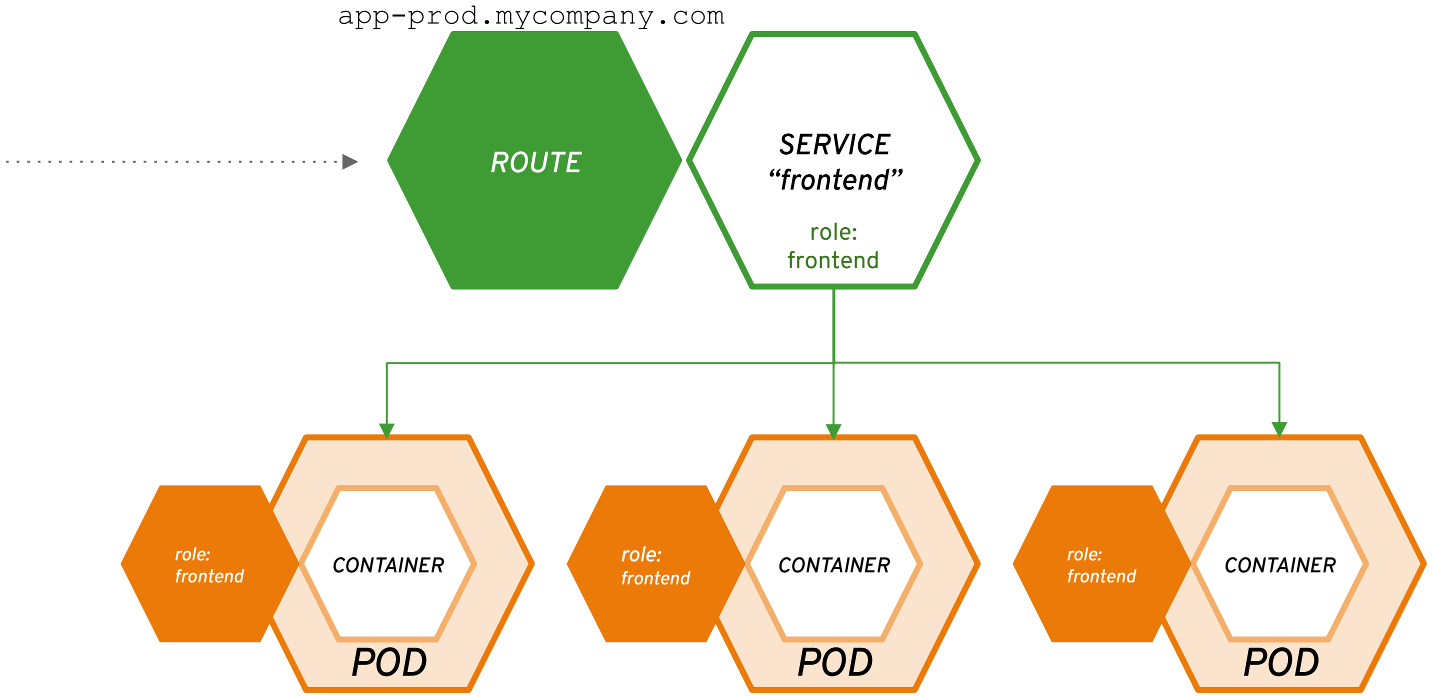
apps can talk to each other via services



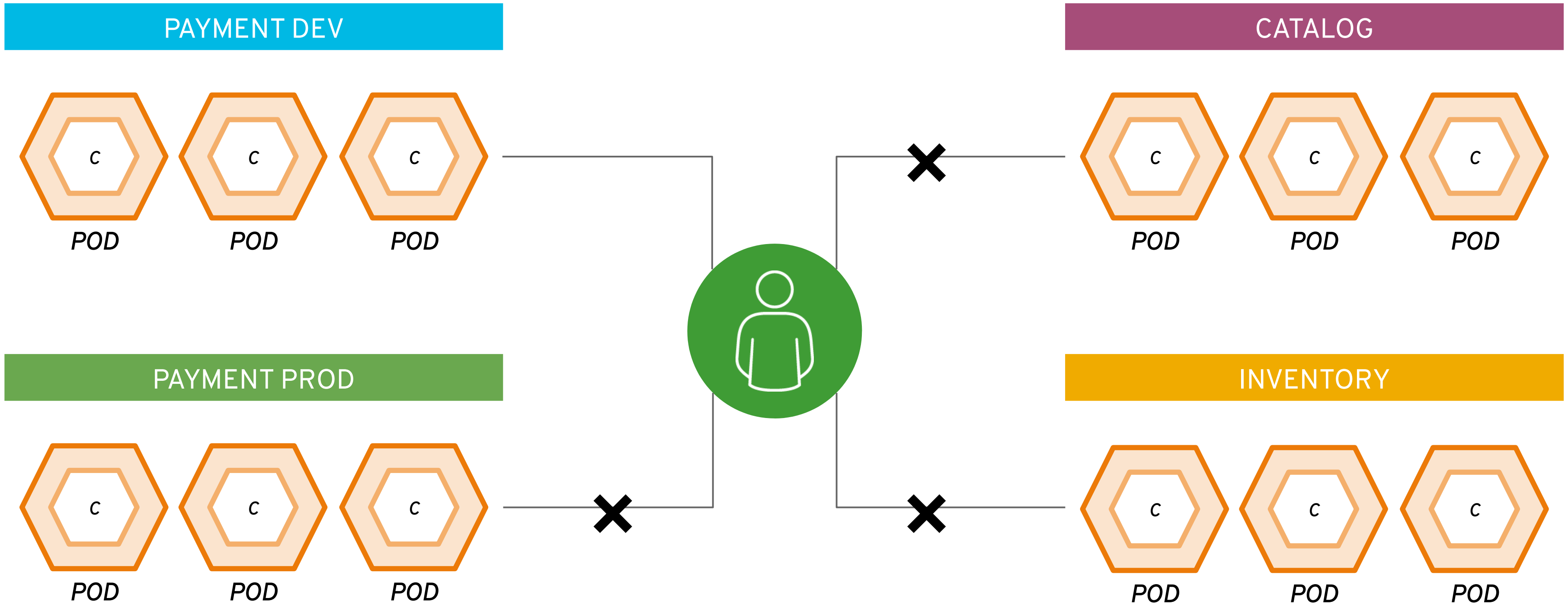
`routes` make services accessible to clients outside the environment via real-world urls



```
> curl http://app-prod.mycompany.com
```



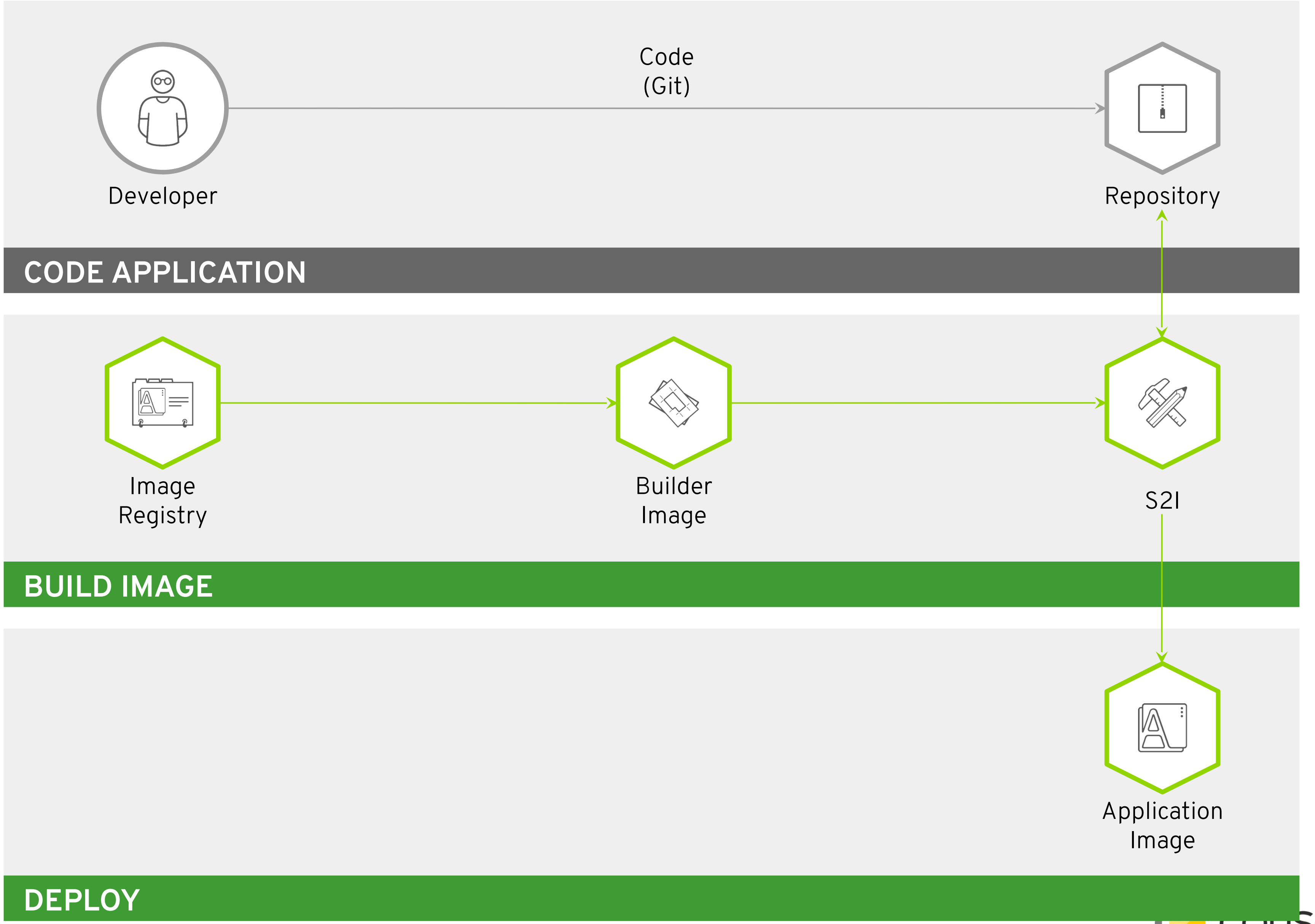
projects isolate apps across environments, teams, groups and departments



DEPLOY YOUR SOURCE CODE

DEPLOY YOUR APP BINARY

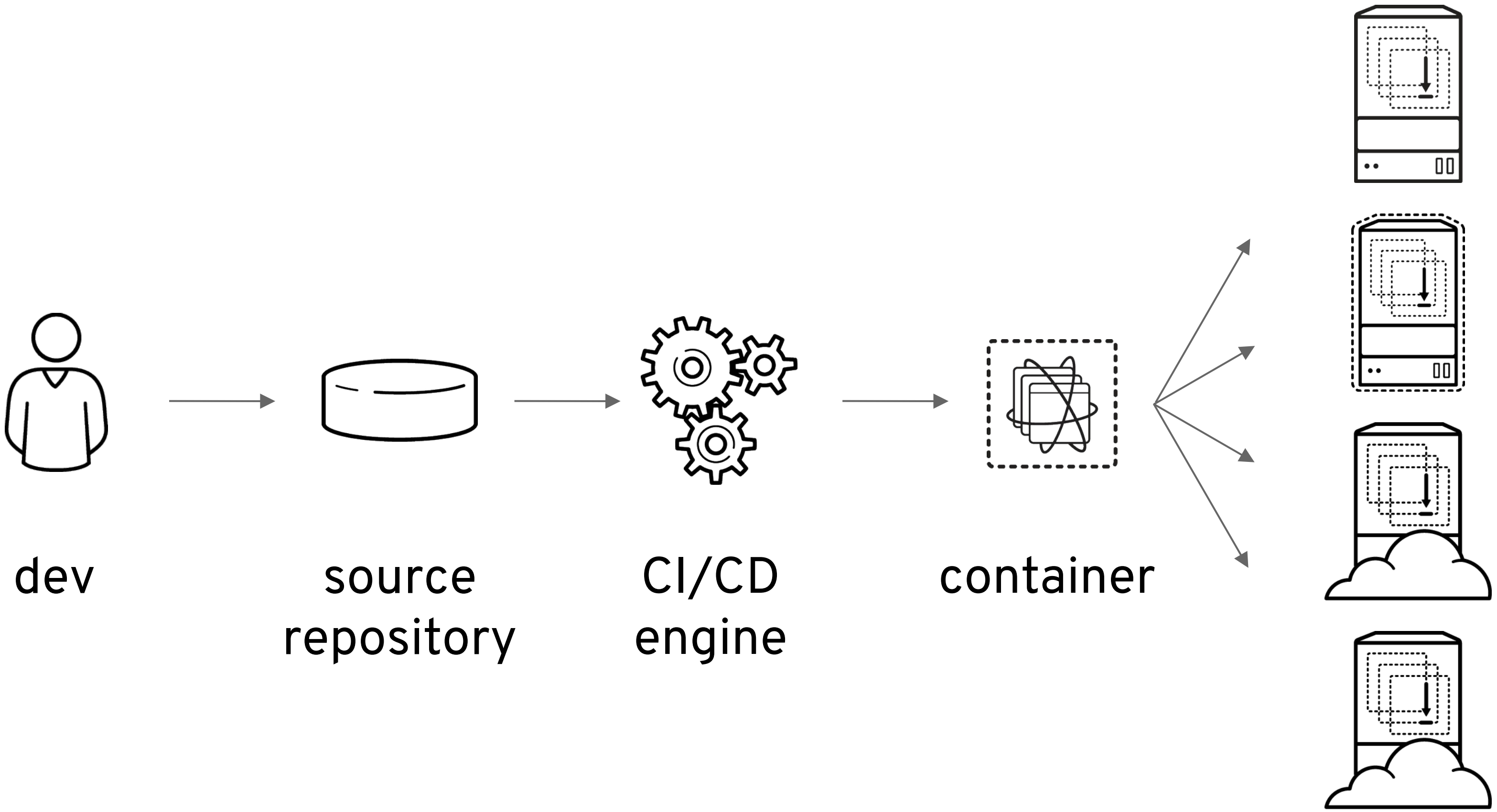
DEPLOY YOUR CONTAINER IMAGE


Code
(Git)
Developer
Repository
CODE APPLICATION
Image
Registry
Builder
Image
S2I
BUILD IMAGE
Application
Image
DEPLOY

Module: Optimizing Existing Applications

Module: Debugging, Monitoring and Continuous Delivery

DEPLOYMENT PIPELINES



dev

source repository

CI/CD engine

container

physical

virtual

private cloud

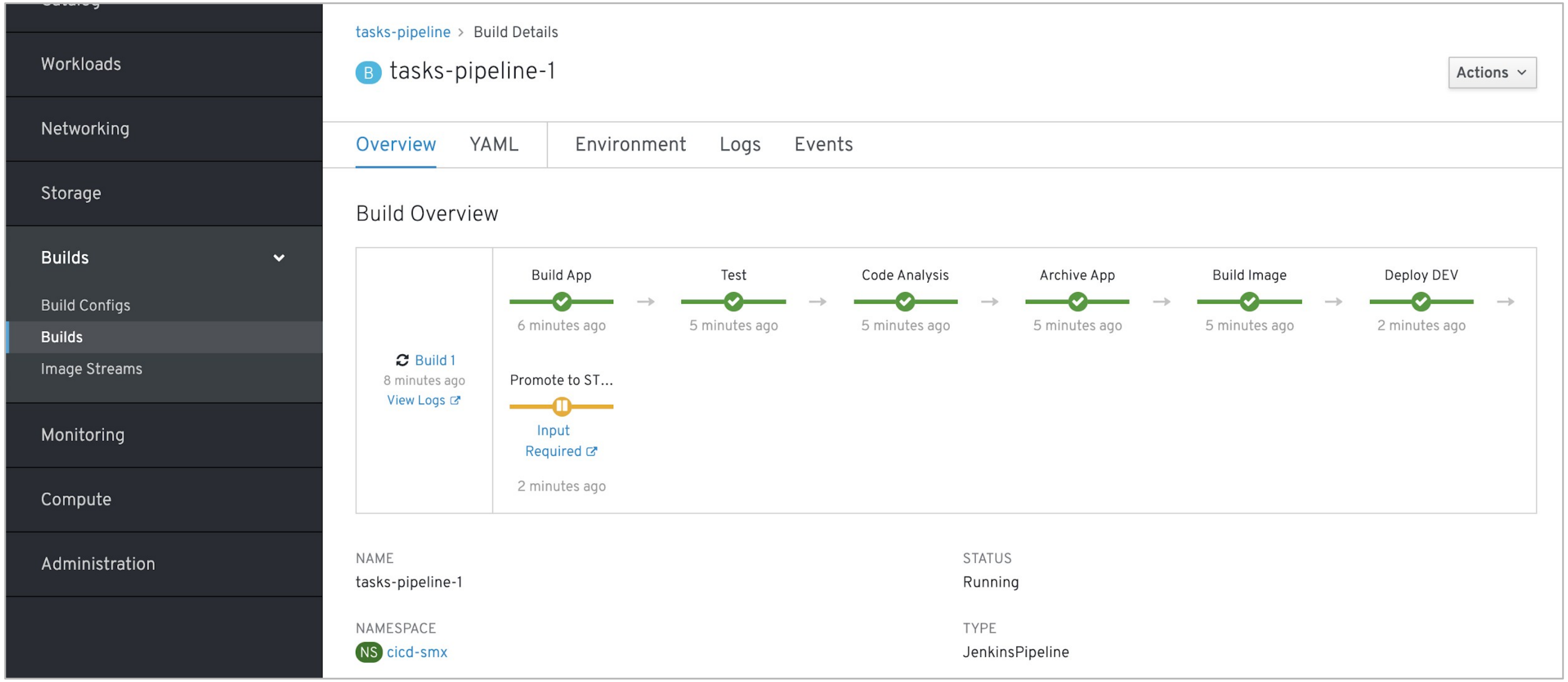
public cloud

OPENSHIFT PIPELINES

Jenkins is still the most used CI/CD platform in enterprises and can be used from inside OpenShift.

An intuitive pipeline visualization makes it simple for users to see how builds are progressing.

The full Jenkins UI is also available.



Build Overview

Build 1
8 minutes ago
View Logs

Build App – 6 minutes ago → Test – 5 minutes ago → Code Analysis – 5 minutes ago → Archive App – 5 minutes ago → Build Image – 5 minutes ago → Deploy DEV – 2 minutes ago →

Promote to ST...
Input Required
2 minutes ago

NAME
tasks-pipeline-1

STATUS
Running

NAMESPACE
NS cicd-smx

TYPE
JenkinsPipeline

OPENSHIFT PIPELINES

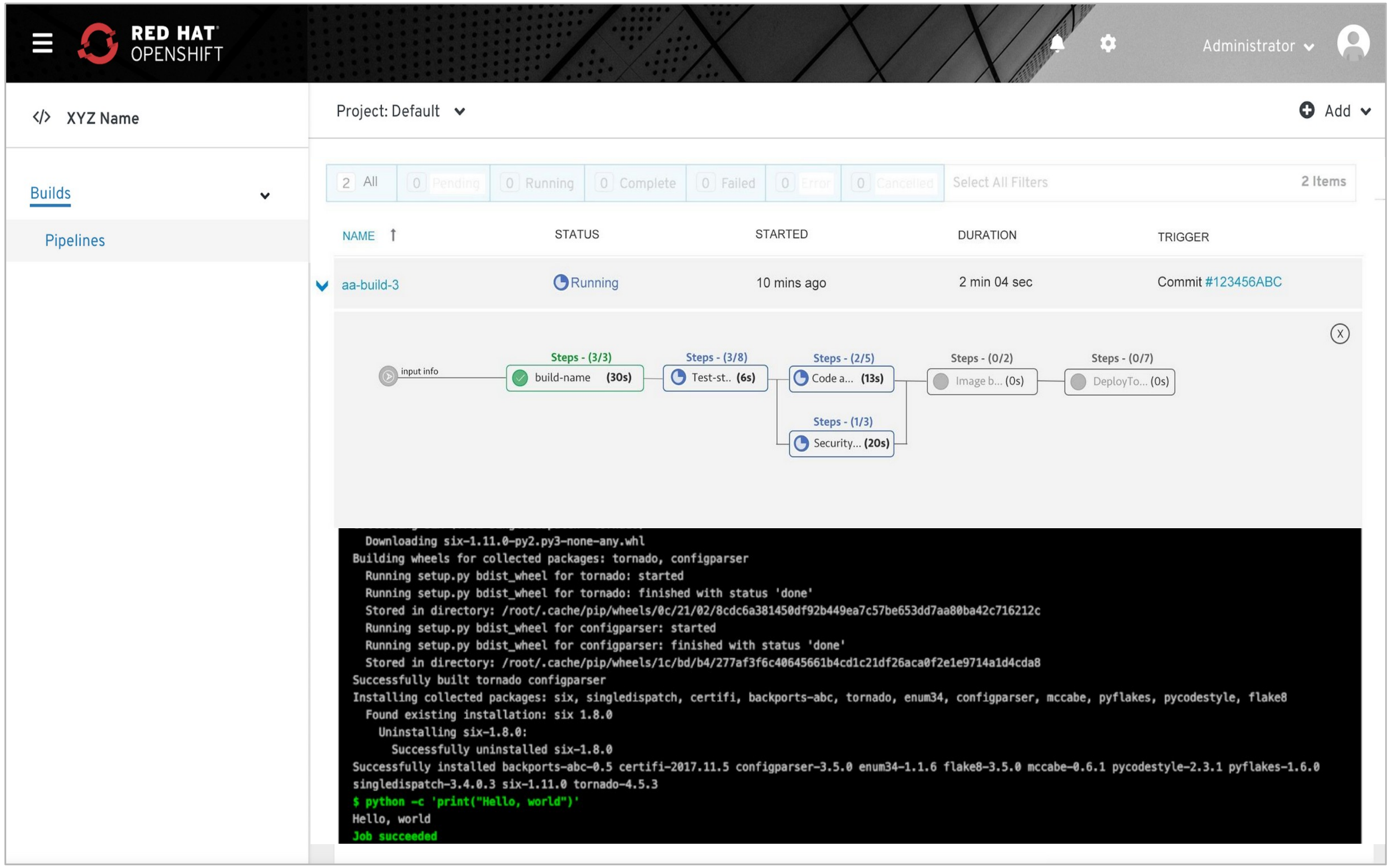
- CI/CD workflow via **Jenkins**
- Pipelines are started, monitored, and managed similar to other builds
- Auto-provisioning of Jenkins server
- On-demand Jenkins slaves
- Embedded Jenkinsfile or in Git repo

```
pipeline {
 agent {
  label 'maven'
 }
 stages {
  stage('build app') {
   steps {
    git url: 'https://git/app.git'
    sh "mvn package"
   }
  }
  stage('build image') {
   steps {
    script {
     openshift.withCluster() {
      openshift.startBuild("...")
     }
    }
   }
  }
 }
}
```

OPENSHIFT PIPELINES CI/CD PLATFORM

Next-gen Kubernetes CI/CD pipeline that works for containers (including serverless).

Based on the **Tekton** project (which was spun out of the Knative Pipelines project) started by Google, Red Hat and others.



NAME	STATUS	STARTED	DURATION	TRIGGER
aa-build-3	Running	10 mins ago	2 min 04 sec	Commit #123456ABC

```
  Downloading six-1.11.0-py2.py3-none-any.whl
Building wheels for collected packages: tornado, configparser
  Running setup.py bdist_wheel for tornado: started
  Running setup.py bdist_wheel for tornado: finished with status 'done'
  Stored in directory: /root/.cache/pip/wheels/0c/21/02/8cdc6a381450df92b449ea7c57be653dd7aa80ba42c716212c
  Running setup.py bdist_wheel for configparser: started
  Running setup.py bdist_wheel for configparser: finished with status 'done'
  Stored in directory: /root/.cache/pip/wheels/1c/bd/b4/277af3f6c40645661b4cd1c21df26aca0f2e1e9714a1d4cda8
Successfully built tornado configparser
Installing collected packages: six, singledispatch, certifi, backports-abc, tornado, enum34, configparser, mccabe, pyflakes, pycodestyle, flake8
  Found existing installation: six 1.8.0
    Uninstalling six-1.8.0:
      Successfully uninstalled six-1.8.0
Successfully installed backports-abc-0.5 certifi-2017.11.5 configparser-3.5.0 enum34-1.1.6 flake8-3.5.0 mccabe-0.6.1 pycodestyle-2.3.1 pyflakes-1.6.0
singledispatch-3.4.0.3 six-1.11.0 tornado-4.5.3
$ python -c 'print("Hello, world")'
Hello, world
Job succeeded
```

OpenShift Application Monitoring

Metrics collection and storage via Prometheus, an open-source monitoring system time series database.

Alerting/notification via Prometheus’ Alertmanager, an open-source tool that handles alerts send by Prometheus.

Metrics visualization via Grafana, the leading metrics visualization technology.

LAB INSTRUCTIONS

- **Everything is done in browser -** no local commands or installs needed on your laptop
- Tested with **Chrome 75.0.3770.142**, **Firefox 60.8.0esr.** → **Safari 12.x does not work!**
- If things get weird, just reload browser page
- **Turn off VPN** (we use websockets extensively), **pause AdBlock** for the lab domain (there are no ads)
- To recreate the lab locally, visit github.com/RedHat-Middleware-Workshops/cloud-native-workshop-v2-infra
- Everyone should have their own **unique logins,** e.g.: user45 / r3dh4t1!

Get Started at: www.consol.de/roadshow

Credentials: userXX / r3dh4t1!

If you get stuck, raise hand

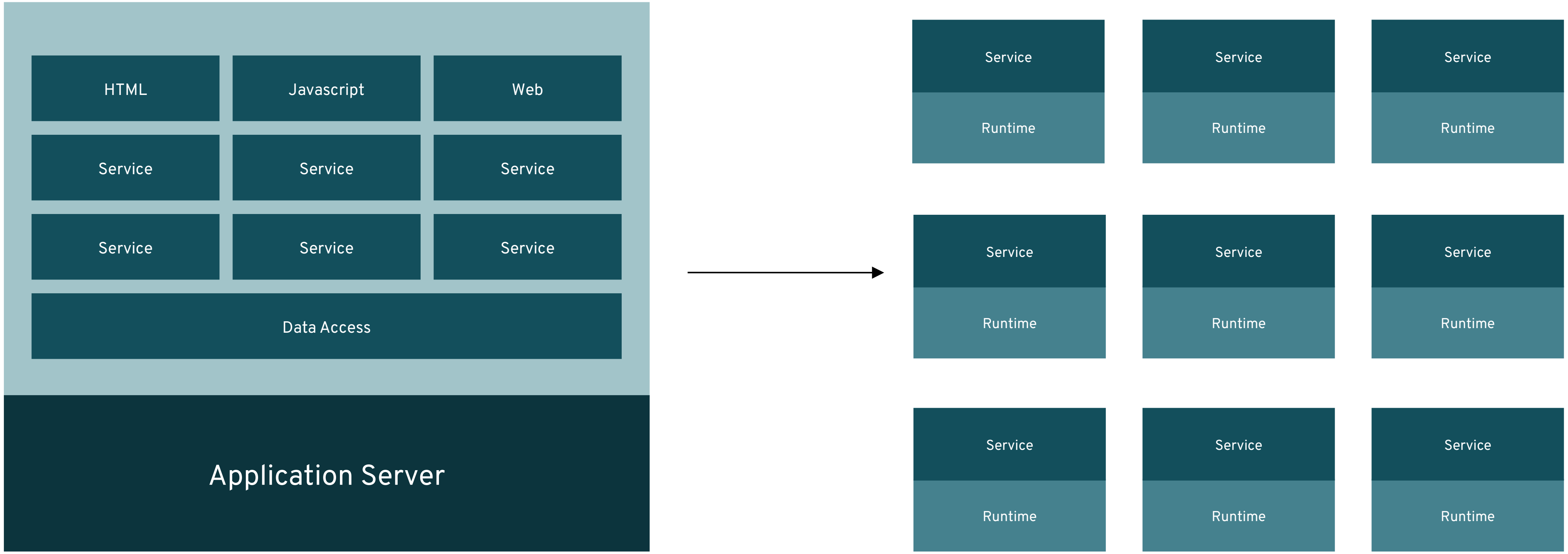
consol Wir unternehmen IT

Red Hat

Control Cloud Native Apps With Service Mesh

~~MICROSERVICES~~ ARCHITECTURE
DISTRIBUTED

CONFIDENTIAL Designator



consol Wir unternehmen IT

Red Hat

DISTRIBUTED COMPUTING CHALLENGES

Fallacies of Distributed Computing

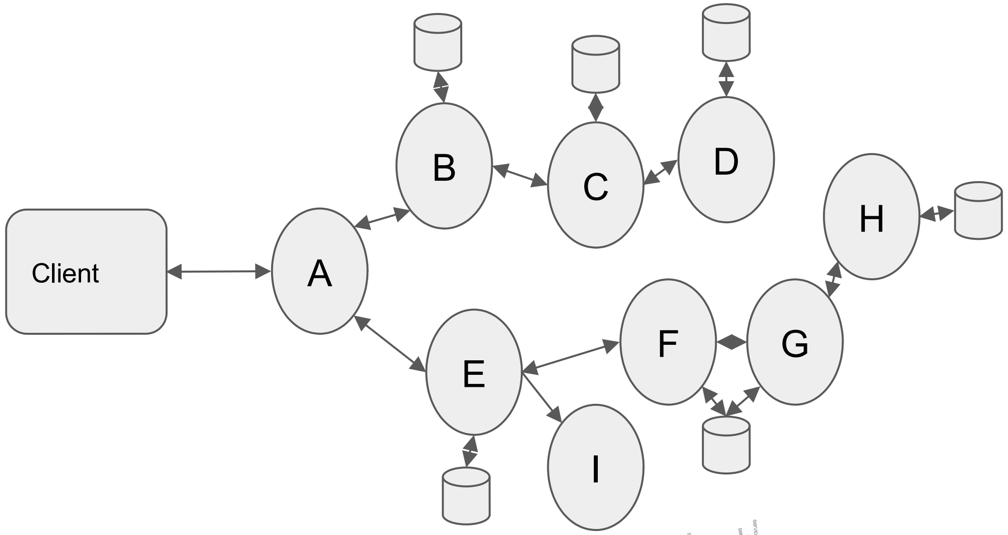
- The network is reliable.
- Latency is zero.
- Bandwidth is infinite.
- The network is secure.
- Topology doesn't change.
- There is one administrator.
- Transport cost is zero.
- The network is homogeneous.

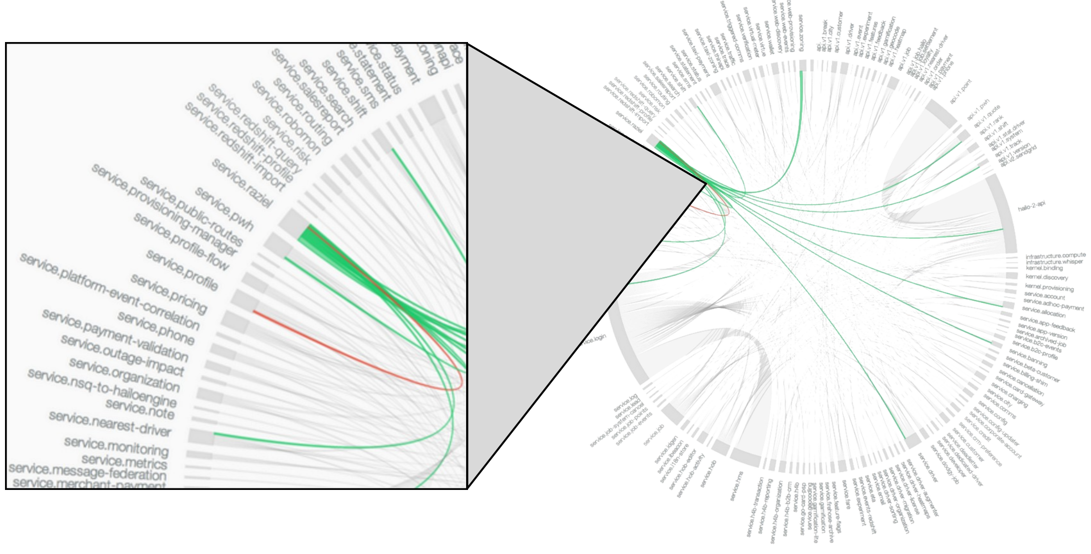
wikipedia.org/wiki/Fallacies_of_distributed_computing

MICROSERVICES ARE HARD

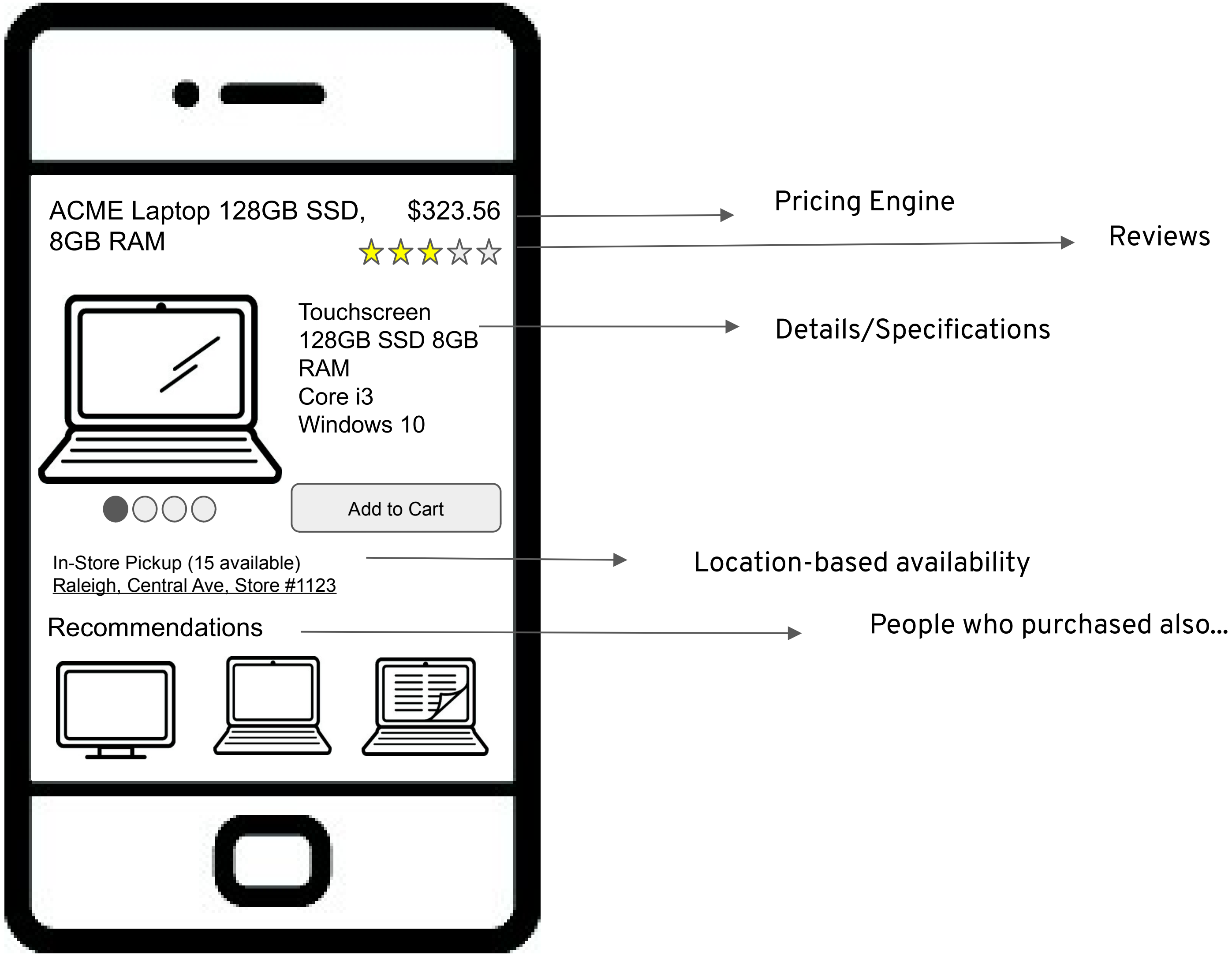
Because applications must deal with

- Unpredictable failures
- End-to-end application correctness
- System degradation
- Topology changes
- Elastic/ephemeral/transient resources
- Distributed logs
- The fallacies of distributed computing





AN EXAMPLE



ACME Laptop 128GB SSD, 8GB RAM

$323.56

★★★☆☆

Touchscreen
128GB SSD 8GB RAM
Core i3
Windows 10

●○○○

Add to Cart

In-Store Pickup (15 available)
Raleigh, Central Ave, Store #1123

Recommendations

Pricing Engine

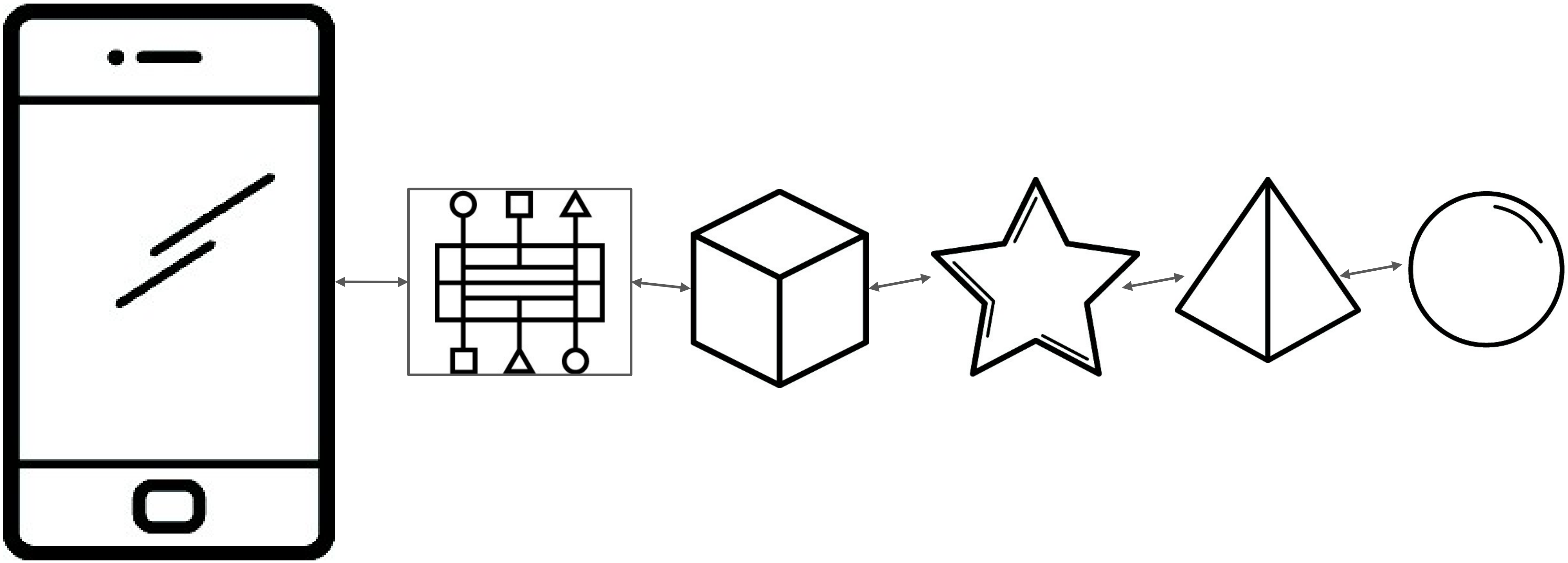
Reviews

Details/Specifications

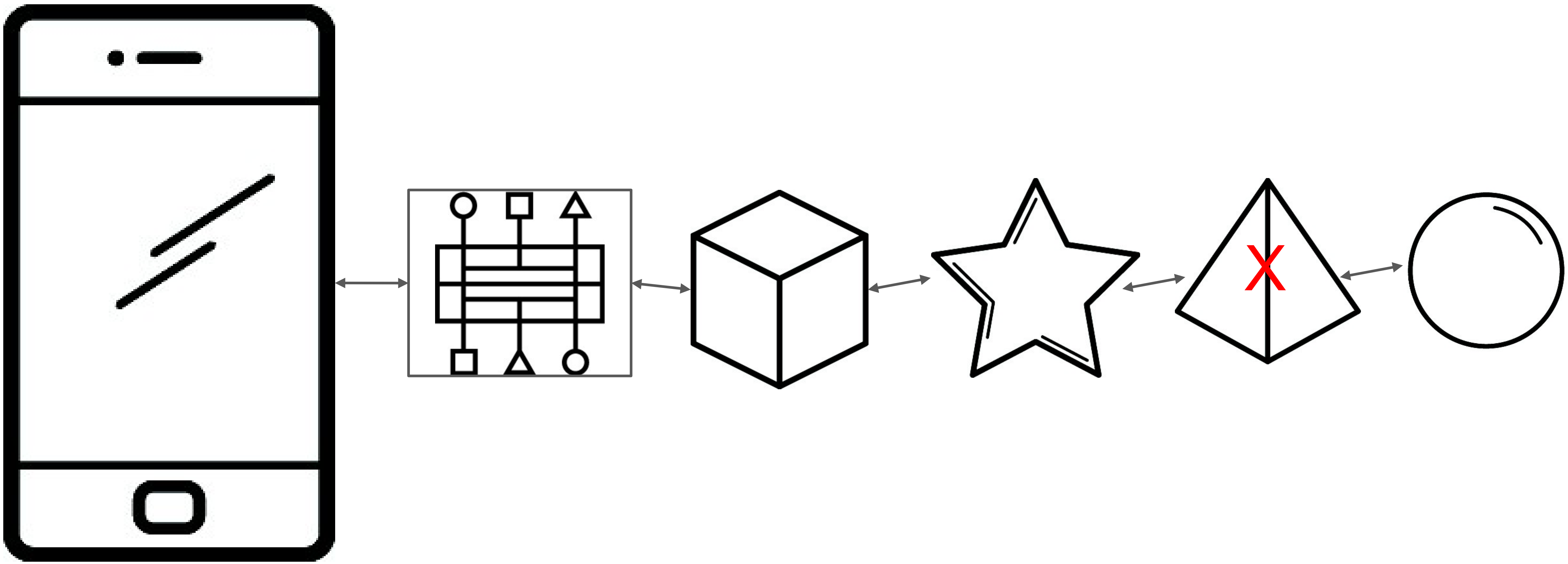
Location-based availability

People who purchased also...

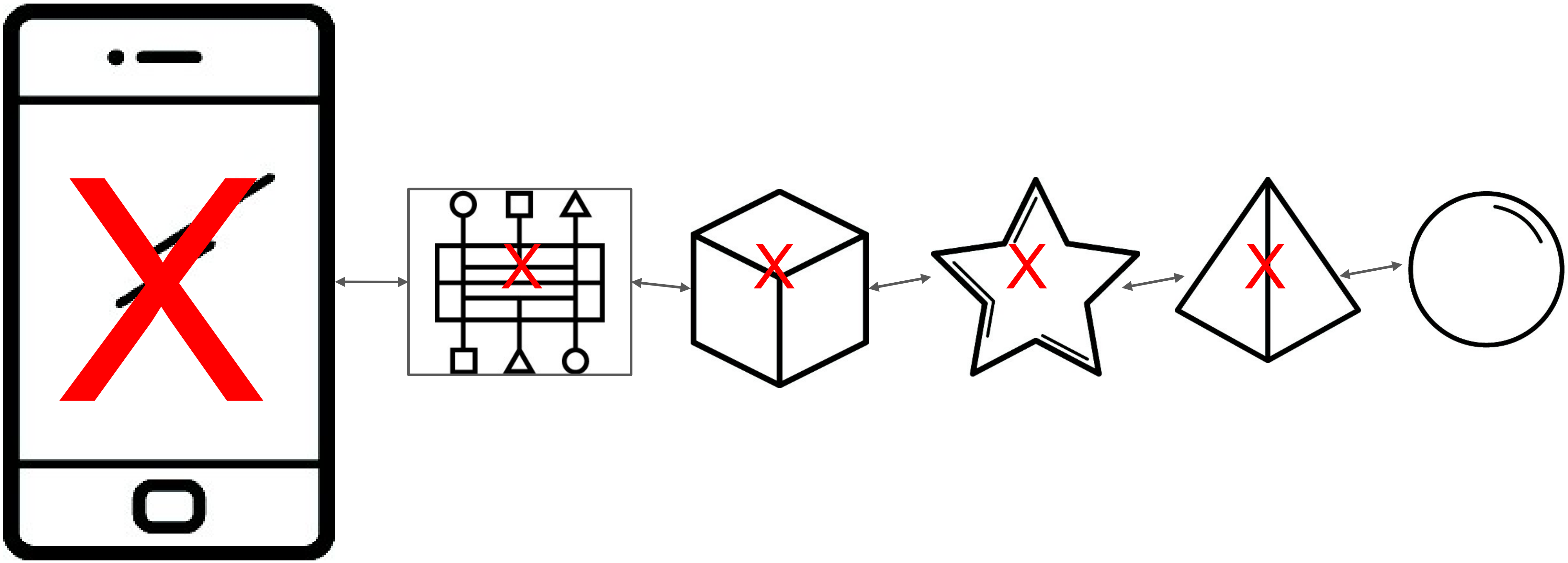
CHAINING



CHAINING (FAILURE)



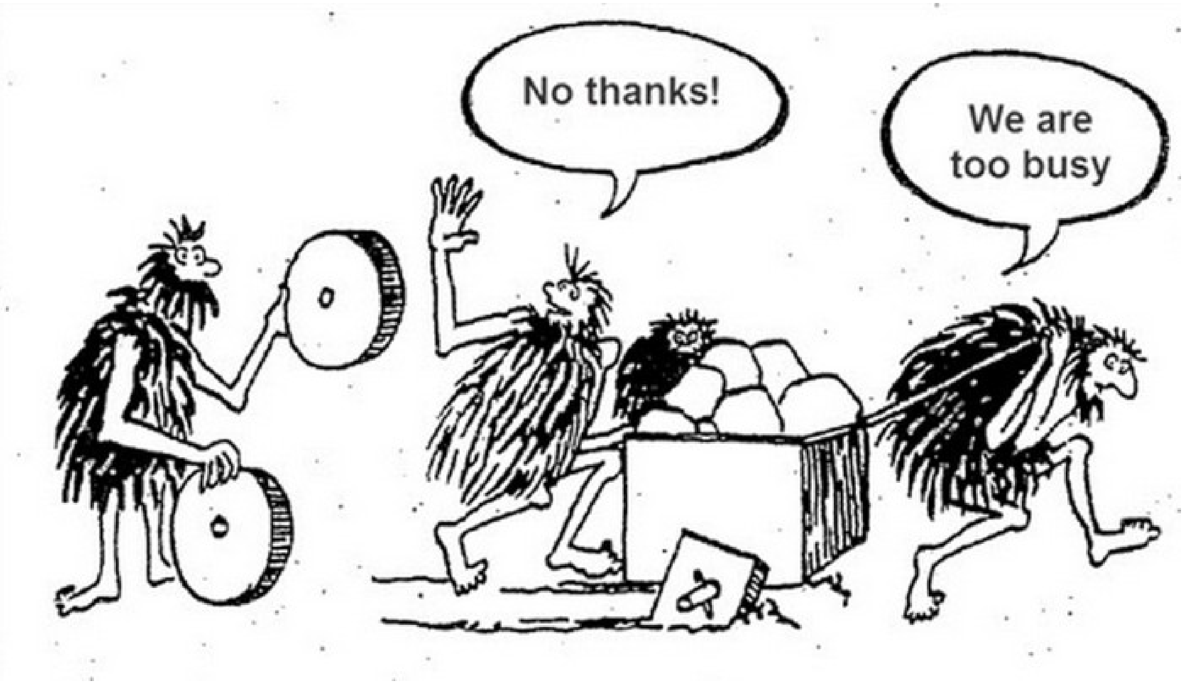
CHAINING (CASCADING FAILURE)



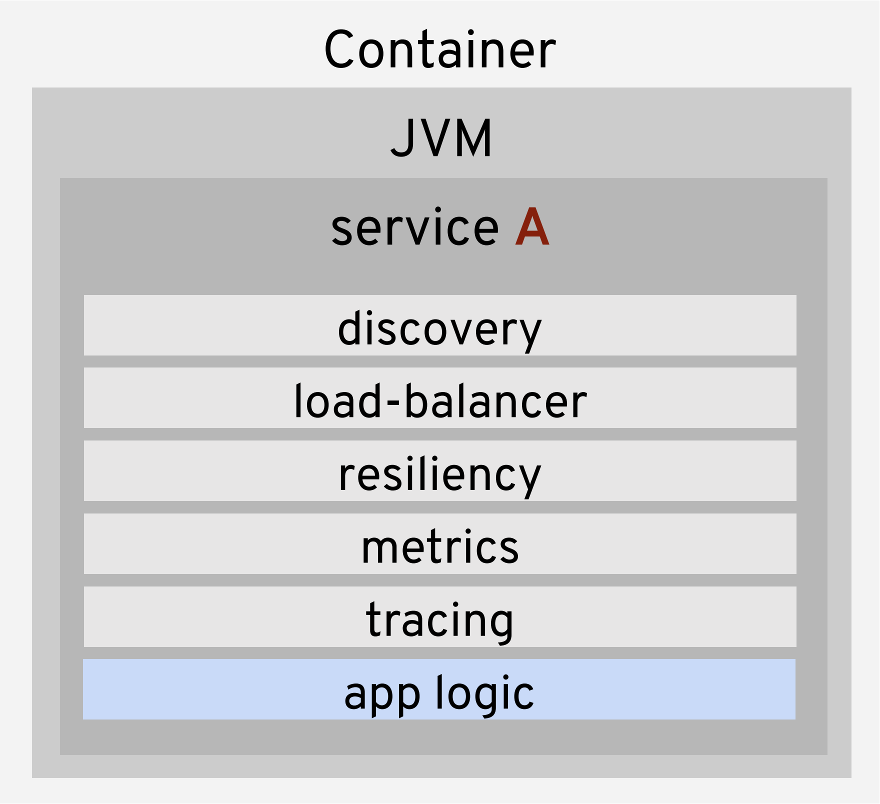
POSSIBLE SOLUTIONS

Have your developers do this:

- Circuit Breaking
- Bulkheading
- Timeouts/Retries
- Service Discovery
- Load Balancing
- Traffic Control



NETFLIX OSS



Hystrix

Zuul

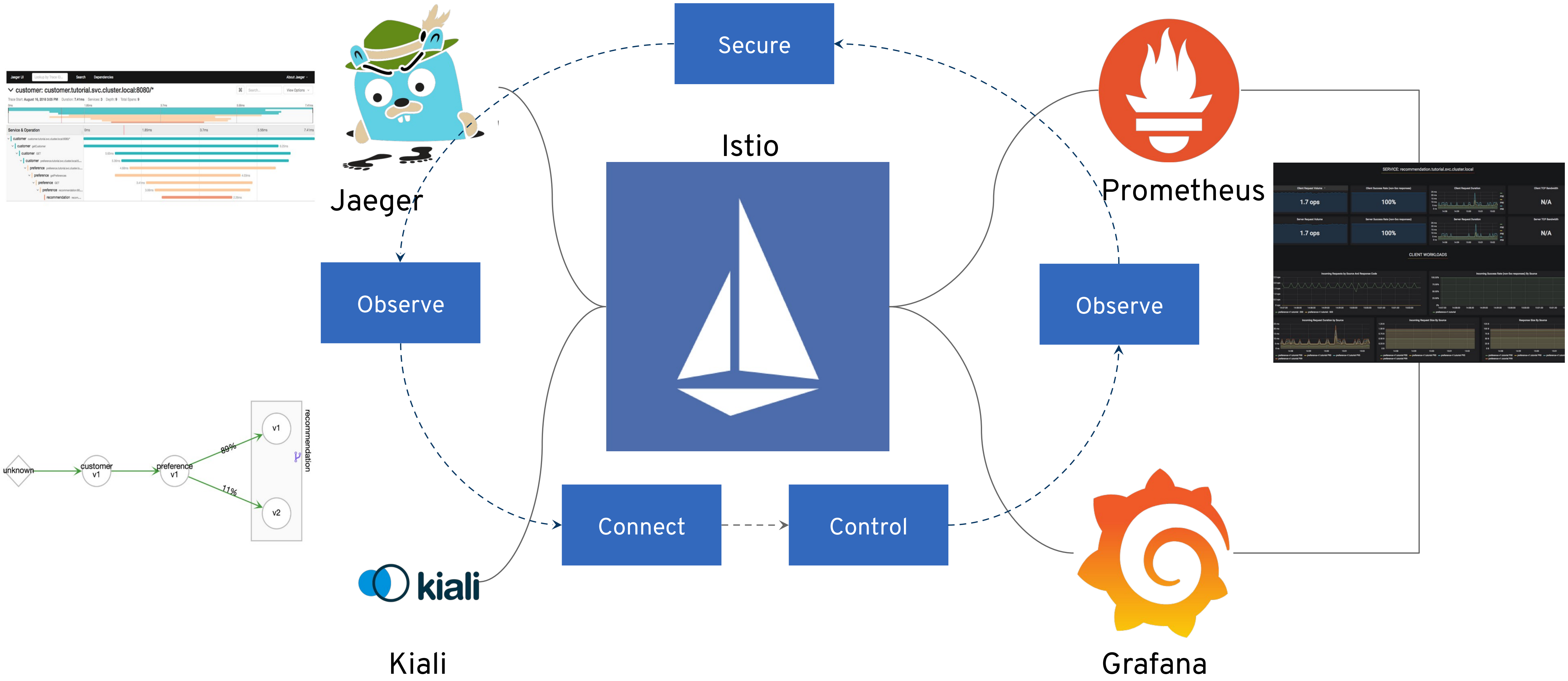
Ribbon

Eureka

Archaius

Need a library to support each language/framework combination

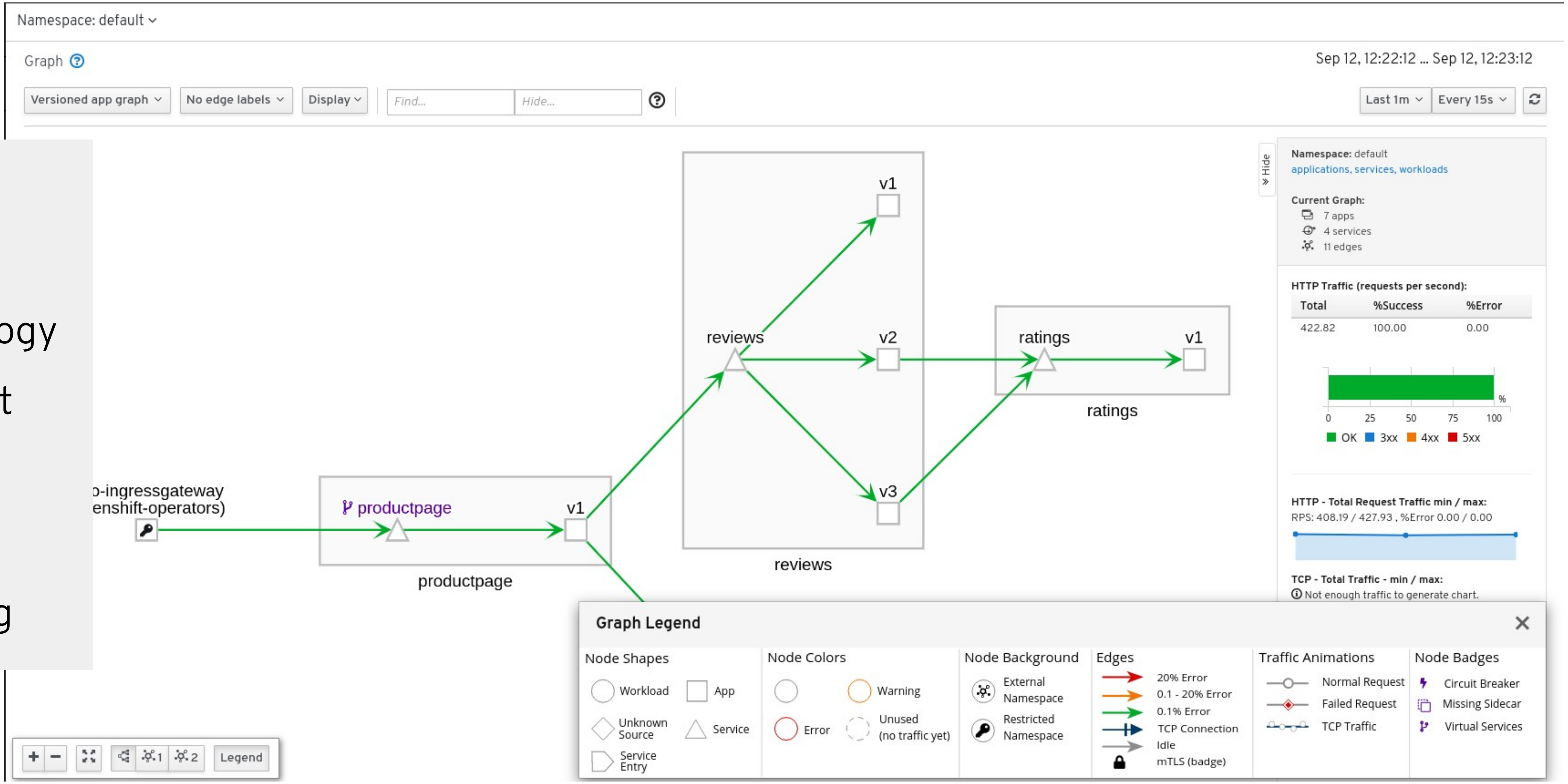
Service Mesh Ecosystem



Enhanced Visualization of Cluster Traffic With Kiali

Visualization of what Matters most:

- Application Topology
- Traffic throughput
- Error Rates
- Service Latency
- Service Versioning



Namespace: default

Graph

Versioned app graph | No edge labels | Display | Find... | Hide...

Sep 12, 12:22:12 ... Sep 12, 12:23:12

Last 1m | Every 15s

Namespace: default
applications, services, workloads

Current Graph:
7 apps
4 services
11 edges

HTTP Traffic (requests per second):

Total	%Success	%Error
422.82	100.00	0.00

HTTP - Total Request Traffic min / max:
RPS: 408.19 / 427.93 , %Error 0.00 / 0.00

TCP - Total Traffic - min / max:
Not enough traffic to generate chart.

consol
Wir unternehmen IT

Red Hat

GOAL FOR LAB

In this lab you will learn:

- How to deploy apps into the **OpenShift Service Mesh**
- How to generate and visualize **deep metrics** for apps with **Kiali** console
- How to **alter routing** dynamically
- How to **inject faults** for testing
- How to do **rate limiting**
- How the mesh implements **circuit breaking** and **distributed tracing**

LAB INSTRUCTIONS

- **Everything is done in browser -** no local commands or installs needed on your laptop
- Tested with **Chrome 75.0.3770.142**, **Firefox 60.8.0esr. → Safari 12.x does not work!**
- If things get weird, just reload browser page
- **Turn off VPN** (we use websockets extensively), **pause AdBlock** for the lab domain (there are no ads)
- To recreate the lab locally, visit github.com/RedHat-Middleware-Workshops/cloud-native-workshop-v2-infra
- Everyone should have their own **unique logins,** e.g.: user45 / r3dh4t1!

Get Started at: www.consol.de/roadshow

Credentials: userXX / r3dh4t1!

If you get stuck, raise hand

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat